In [11]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from pandas.testing import assert_frame_equal
```

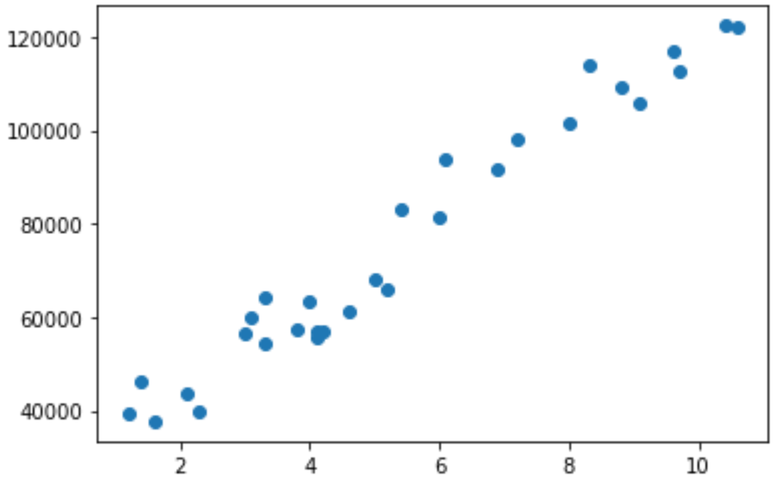
In [60]:

```
data=pd.read_csv('/Users/diyaddin/Documents/Salary_dataset.csv')

veri=data.copy()

y=veri['Salary']
X=veri['YearsExperience']

plt.scatter(X,y)
plt.show()
```



In [19]:

```
sabit=sm.add_constant(X)
model=sm.OLS(y,sabit).fit()
print(model.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                 Salary   R-squared:                       0.95
7
Model:                            OLS   Adj. R-squared:                  0.95
5
Method:                 Least Squares   F-statistic:                     622.
5
Date:                Mon, 27 Nov 2023   Prob (F-statistic):           1.14e-2
0
Time:                        06:03:03   Log-Likelihood:                -301.4
4
No. Observations:                  30   AIC:                             606.
9
Df Residuals:                      28   BIC:                             609.
7
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
======
                      coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
------
const            2.485e+04   2306.654     10.772      0.000    2.01e+04     2.
96e+04
YearsExperience  9449.9623    378.755     24.950      0.000    8674.119     1.
02e+04
```

```
=============================================================================
=
Omnibus:                        2.140   Durbin-Watson:                   1.64
8
Prob(Omnibus):                  0.343   Jarque-Bera (JB):                1.56
9
Skew:                           0.363   Prob(JB):                        0.45
6
Kurtosis:                       2.147   Cond. No.                         13.
6
=============================================================================
=

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [20]:

```python
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

In [24]:

```python
lr.fit(X.values.reshape(-1,1) ,y.values.reshape(-1,1))
print(lr.intercept_, lr.coef_)
```

```
[24848.20396652] [[9449.96232146]]
```

In [26]:

```python
print(lr.predict(X.values.reshape(-1,1) ))
```

```
[[ 36188.15875227]
 [ 38078.15121656]
 [ 39968.14368085]
 [ 44693.12484158]
 [ 46583.11730587]
 [ 53198.09093089]
 [ 54143.08716303]
 [ 56033.07962732]
 [ 56033.07962732]
 [ 60758.06078805]
 [ 62648.05325234]
 [ 63593.04948449]
 [ 63593.04948449]
 [ 64538.04571663]
 [ 68318.03064522]
 [ 72098.0155738 ]
 [ 73988.00803809]
 [ 75878.00050238]
 [ 81547.97789525]
 [ 82492.9741274 ]
 [ 90052.94398456]
 [ 92887.932681  ]
 [100447.90253816]
 [103282.8912346 ]
 [108007.87239533]
 [110842.86109176]
 [115567.84225249]
 [116512.83848464]
 [123127.81210966]
 [125017.80457395]]
```

Our predict function

Y=24848.20396652+9449.96232146*X

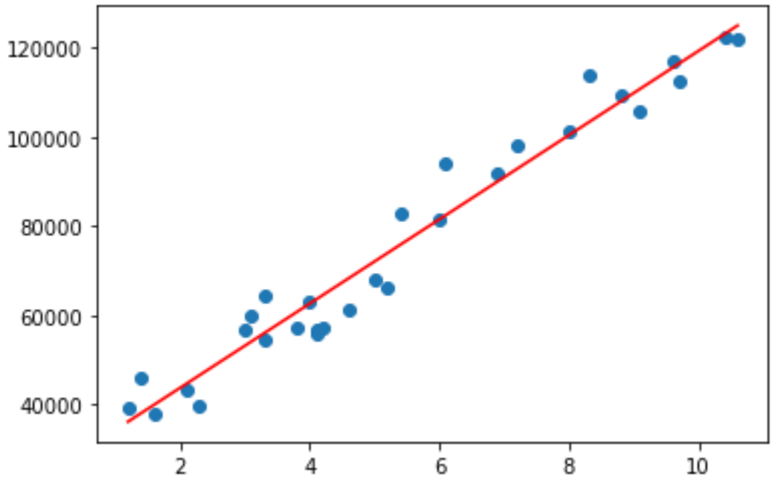
In [61]:

```python
plt.scatter(X,y)

plt.plot(X,24848+9449*X , color='red')
```

Out[61]:

```
[<matplotlib.lines.Line2D at 0x7f83786c1670>]
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: